Control of Software via Bundling

Inventors: Mark Smith and John Boldon

ATTORNEY'S DOCKET NO. 10012134-1

CONTROL OF SOFTWARE VIA BINDLING

TECHNICAL FIELD

The subject matter disclosed herein relates to control of software for multifunctional devices, such as, but not limited to, multifunctional printers.

BACKGROUND

A need exists for methods and systems for controlling software, in particular, revisions to software on devices such as, but not limited to, printers.

SUMMARY

Methods and systems for developing, loading and/or revising software on multifunctional devices, including multifunctional printers and multifunctional devices for receiving bundled software.

BRIEF DESCRIPTION OF THE DRAWINGS

Various methods and/or devices are illustrated by way of example and not limitation in the figures of the accompanying drawings. The same numbers are used throughout the figures to reference like components and/or features.

Fig. 1 illustrates a network environment in which multiple servers, workstations, and printers are coupled to one another via a data communication network.

Fig. 2 is a block diagram showing pertinent components of a printer suitable for use with various exemplary systems and/or methods described herein.

Fig. 3 is a block diagram showing pertinent components of a computer workstation suitable for use with various exemplary systems and/or methods described herein.

Fig. 4 is a block diagram of an exemplary multifunctional printer showing various functional blocks.

Fig. 5 is a block diagram showing a group of printers having at least one wired or wireless communication link.

Fig. 6 is a functional block diagram showing various sub-blocks.

Fig. 7 is a block diagram illustrating an exemplary method for revising software.

Fig. 8 is a block diagram illustrating an exemplary method for revising software.

DETAILED DESCRIPTION

Fig. 1 illustrates a network environment in which multiple servers, workstations, and printers are coupled to one another via a data communication network 100. The network 100 couples together servers 102, 104, computer workstations 106, 108, and printers 110, 112. The network 100 can be any type of network, such as a local area network (LAN) or a wide area network (WAN), using any type of network topology and any network communication protocol. The network may include wired, wireless or a combination of wired and

wireless links. In a particular embodiment, the network 100 is the Internet. Although only a few devices are shown coupled to the network 100, a typical network may include tens or hundreds of devices coupled to one another. Furthermore, the network 100 may be coupled to one or more other networks, thereby providing coupling between a greater number of devices.

The servers 102, 104 may be file servers, email servers, database servers, print servers, or any other type of network server. The workstations 106, 108 can be any type of computing device, such as, but not limited to, a personal computer. Particular exemplary methods and/or systems include printers (e.g., such as the printers 110, 112) that are laser printers. Alternative exemplary methods and/or systems include, for example, ink-jet, bubble-jet or, in general, any type of printer. Furthermore, teachings presented herein optionally apply to any type of printing device, such as scanners, copiers and fax machines. Details of multifunctional devices and/or multifunctional printers appear below; some details of such devices and/or printers are known in the art. Although not shown in Fig. 1, one or more workstations and/or servers may contain a print rendering engine capable of converting raw print job information into a particular format (e.g., language) understood by certain types of printers. Where a system includes a printer menu, a printer menu editor application is optionally executed on a workstation 106, 108, or on a server 102, 104, to create or modify a printer menu structure.

Fig. 2 is a block diagram showing pertinent components of a printer 110 suitable (e.g., printer 110, 112 of Fig. 1) for use with various examples presented herein. As shown in Fig. 2, the printer 110 includes a processor 120, an electrically erasable programmable read-only memory (EEPROM) 122, and

a random access memory (RAM) 124. The processor 120 processes various instructions necessary to operate the printer 110 and/or communicate with other devices. The EEPROM 122 and the RAM 124 store various information such as, but not limited to, configuration information, fonts, templates, data being printed, and menu structure information. Although not shown in Fig. 2, a particular printer may also contain a ROM (non-erasable) in place of or in addition to the EEPROM 122.

The exemplary printer 110, as shown in Fig. 2, also includes a disk drive 126, a network interface 128, and a serial/parallel interface 130. The disk drive 126 provides additional storage for data being printed or other information used by the printer 110. Although both the RAM 124 and the disk drive 126 are illustrated in Fig. 2, a particular printer may contain either a RAM 124 or a disk drive 126, depending on the storage needs of the printer. For example, an inexpensive printer may contain a small amount of RAM 124 and no disk drive 126, thereby reducing the manufacturing cost of the printer. The network interface 128 provides a connection between the printer 110 and a data communication network, such as the network 100. The network interface 128 allows devices coupled to a common data communication network to send print jobs, menu data, and other information to the printer 110 via a network (e.g., network 100 of Fig. 1). Similarly, the serial/parallel interface 130 provides a data communication path directly between the printer 110 and another device, such as a workstation, server, or other computing device. Although the printer 110 shown in Fig. 2 has two interfaces (the network interface 128 and the serial/parallel interface 130), an exemplary printer may include one interface or more than two interfaces.

As shown in Fig. 2, the exemplary printer 110 also contains a user interface/menu browser 132 and a display panel 134. The user interface 132 may include a series of buttons, switches or other indicators that are optionally usable by a user of the printer 110. The display panel 134 includes a graphical display that typically provides information regarding the status of the printer and the current options available through a menu structure. The printer 110 display panel 134 may display various menu options to a user of the printer 110. The display panel 134 and associated control buttons optionally allow a user of the printer to navigate a printer's menu structure.

Fig. 3 is a block diagram showing pertinent components of a computer workstation 106 in accordance with exemplary methods and/or systems described herein. The workstation 106 includes a processor 140, a memory 142 (such as ROM and RAM), user input devices 144, a disk drive 146, interfaces 148 for inputting and outputting data, a floppy disk drive 150, and a CD-ROM drive 152. The processor 140 performs various instructions to control the operation of the workstation 106. The memory 142, the disk drive 46, the floppy disk drive 150, and the CD-ROM drive 152 provide data storage mechanisms. The user input devices 144 include a keyboard, mouse, pointing device, or other mechanism for inputting information to the workstation 106. The interfaces 148 provide a mechanism for the workstation 106 to communicate with other devices. A server (e.g., servers 102, 104 of Fig. 1), typically includes at least some of the workstation 106 features.

Various printers described herein have printer specific software (or firmware) and software (or firmware) for all paper handling and/or print associated devices (e.g., scanners, finishing units, etc.). Printers having

multiple functions are sometimes referred to herein as multifunctional printers (MFPs) which are within the class of multifunctional devices (MFDs). As the name implies, a MFD is a device capable of performing multiple functions. Regarding printers, most functions relate to printing, paper handling, and/or data communication. Examples of the functions performed by a MFD include, but are not limited to: printing; multi-orig-out-ing (sometimes referred to, or including, mopy printing); copying; sending/receiving fax; sending/receiving e-mail; image acquisition; text recognition; source paper handling (such as high capacity input (HCI), trays having differing paper sizes/types, envelope trays, etc.); destination paper handling (such as stacking, stapling, sorting, collating, mailboxes, etc.); and data acquisition (network, serial, USB, wireless, parallel, floppy disk, hard disk, RAM, flash cards, etc.).

An exemplary function, as listed above, is mopy printing. Mopy-enabled printers utilize intelligent software (or firmware) and host software to improve throughput and reduce or eliminate various issues. In mopy printing, a printer driver typically transmits a file only once and sends appropriate header information (e.g., the number of copies) to the printer for rasterization and spooling. In general, the printer rasterizes the job only once and then prints multiple, original copies in the manner determined by a user. Mopy printing is particularly useful for complex jobs that slow down the printer when the first copy is created or takes a long time to transmit over a network.

Commercially available printers having multiple functionality include the Hewlett Packard (HP) LASERJET® 9000 printers (Palo Alto, California). Such MFPs are based on installation of software and/or hardware solutions in new or existing printers. In particular, such MFPs optionally include e-enabled

print management features which may perform the following management tasks: install printers, configure printer options, configure and install multiple printers at one time, troubleshoot printer problems, schedule discoveries to update the device cache during off-peak hours, search for a specific printer or group of printers based on a wide variety of criteria, check a printer's current status, check the status of a printer's consumables, verify the type of media loaded in a printer's trays, verify a printer's capabilities, organize printers by logical groups, and create a virtual office layout with dynamic site maps.

Such MFPs address issues related to the variety and volume of transactions in today's business. In particular, an MFP may eliminate the need for a dedicated specialized high volume printer for each type of transaction document. As referred to herein, the term "transaction printing" is an industry standard term that refers to a scheduled batch print jobs with a copy count of one. Normally these jobs are generated from a mainframe or from an NT® operating system server (Microsoft Corporation, Redmond, Washington) or UNIX® operating system server (UNIX System Laboratories, Inc., Basking Ridge, New Jersey). Typical output examples include invoices, statements, payroll, purchase orders, accounts receivable, etc.

Additional MFP features may include: browser-compatible printer management software having an embedded web server and administration software (e.g, HP WEB JETADMIN®) to deliver remote printer management services and solutions for one-to-one and one-to-many printer management; Internet software that may include easy custom Internet software disk replication utility to customize what options administrators will load to their systems; administration software for easy intranet administration; Internet

installer software for easy downloading and updating of software; software utility software to let a user or an administrator know when updates are available; Internet page set up utility software to format and print internet pages, and to poll the internet for addresses that a user (or device) has requested as well as schedule the printing of those selected pages at any specified interval or intervals.

An MFP may also have an ability to send automatic e-mail alerts, and be optionally configured to send preconfigured notifications of printer problems and job status via e-mail to designated recipients on the network. An e-mail notice may even provide a hot link for ordering supplies via the Internet.

An MFP may further include smart components. For example, a print supplies system may be activated upon installation of a smart print cartridge in a printer. Each smart print cartridge may feature its own chip which may help to measure and report toner levels, provide usage information, assure job completion, etc. In such an MFP, the cartridge, printer, and software optionally (and typically) work cooperatively.

In general, a MFD (e.g., a MFP) includes multiple parts that rely on software. Such parts include, but are not limited to: formatters, paper-handling devices, copy processor boards, etc. These parts operate in conjunction with firmware/software that may be subject to revision or upgrades. As described herein, such upgrades may be performed remotely. Prior to the existence of such MFPs, issues associated with part compatible were virtually nonexistent.

As described herein, an exemplary method of bundled upgrading helps to assure part-to-part operational compatibility. In this method, upgrading of software on a wholly part-by-part basis is disabled and instead, all upgradeable (or revisable) software is upgraded (or revised) through use of a software bundle. According to this method, each software component of the software bundle has a guarantee of known compatibility. For example, when a new software upgrade for a part becomes available it is qualified with existing software upgrades for other parts of a MFP. In an exemplary upgrade method, once a new software upgrade has passed a qualification procedure, a new software bundle is created comprising the new software upgrade (or upgrades) and other qualified existing software.

Software components suitable for inclusion in a software bundle optionally include a "printer driver", a software program that enables other programs to work with a particular printer without concern for specifics of the printer's hardware and internal language. In general, a printer requires a specific set of codes and commands to operate properly and to provide access to special features and abilities. Other multifunctional devices generally have a device driver (i.e., software component) that provides for similar functionality.

Network administrators within an organization typically use one or more device management applications to manage peripheral devices within an organizational or corporate intranet. A peripheral device is any device that can be connected to a computer or network such as a printer, copier, scanner, fax machine, data storage system, lab equipment, a home entertainment device, and the like. HP JETADMIN® and HP WEB JETADMIN® products are exemplary peripheral device management applications that may be used by

network administrators to discover, install, monitor and troubleshoot network-connected peripherals (or devices), such as printers, in an intranet.

Referring to Fig. 4, a block diagram of an exemplary multifunctional printer 410 is shown (see also printer 110, 112 Fig. 1 and Fig. 2). As shown, printer 410 includes a variety of functional blocks which have associated software. Such software optionally operates in conjunction with features of the printer 110 shown in Fig. 1 and Fig. 2 (e.g., processor 120, RAM 124, etc.).

A communication block 412 includes software for various communication operations. Communication software includes, but is not limited to, Web server software capable of hosting Web pages. Such Web pages optionally provide information on printer status, printer configuration, event logs, supplies, usage, network settings, administrative control, diagnostics, other devices, etc. For example, a device Web page hosted on a printer Web server may allow a user to configure a printer (or printers) from a printer or a computer. Commercially available software includes HP WEB JETADMIN® software, which is a Web browser-based management tool for proactive management and configuration of network-connected peripherals. When embedded in a printer, such software (or an equivalent thereof) may allow for remote installations, configurations, diagnostics, and print management for a plurality of network peripherals from a common Web browser, which is optionally located on a printer. Other software associated with Web browser and/or Web server software includes, but is not limited to, virtual machine software (e.g., HP CHAI® virtual machine, Sun Microsystems, Inc. JAVA® virtual machine, etc.). Additional communication software may perform e-mail operations and the like.

A printing block 414 includes software for various printing operations. For example, the printer 410 may include a duplex unit for duplexing print media, thereby allowing for two-sided printing. Thus, a printing block 414 optionally includes software for duplex operations in addition to software for other printing operations. A finishing block 418 includes software for various finishing operations. For example, the printer 410 may include a finishing unit such as a sheet stacker and/or a sheet stapler/stacker for stacking and/or stapling printed media. Such finishing units may increase printer output capacity, separate printed media (e.g., based on job, etc.), and/or fix (e.g., staple, etc.) selected printed media.

A digitizing block 422 includes software related to digital representation of a print image. Further details of an exemplary digitizing block appear below with reference with to Fig. 6. Other functional blocks, whether or not illustrated in Fig. 4, may also include sub-blocks. A print media block 426 includes software for print media access operations. For example, the printer 410 may include a print media supply unit for supplying print media. A consumables block 430 includes software for monitoring and/or controlling consumables. For example, the printer 410 may include sensors for sensing ink or toner levels. The printer 410 may also include a smart toner cartridge. A smart toner cartridge includes an integrated circuit chip or processor to help in sensing, monitoring, and/or reporting toner levels and/or other information. According to the exemplary printer 410, a smart cartridge operates cooperatively with consumable block 430 software. Other printer-related components (e.g., finishing units, media units, etc.) may also include an

integrated circuit chip or processor; such components are referred to herein as "smart" components.

Referring to Fig. 5, a cluster of printers 500 (510, 510', 510", 510"") is shown. In such an arrangement, one printer (e.g., 510) may download software to other printers (e.g., 510', 510", 510"") or a computer may download software to a plurality of printers on a network. As shown in Fig. 5, the cluster 500 includes a communication link 520, which optionally includes wired and/or wireless technologies. One or more of the printers in the cluster of printers 500 optionally include features of the exemplary printer 410, as shown in Fig. 4.

Referring to Fig. 6, an exemplary digitizing block 422 is shown having various sub-blocks. A control panel/user interface sub-block 460 includes software for managing a user interface and/or general print job control. A copy processor sub-block 462 includes software for generating selected data file formats. A scanner service sub-block 464 includes software for scanner services, such as, but not limited to, a scanner service agent. A digital service sub-block 466 includes software for digital services, such as, but not limited to, a service that allows mobile device users to communicate with a printer without use of a separate server or other host. An export tool sub-block 468 includes software for export of information related to print jobs. For example, an export tool sub-block 468 may include software for updating address books from a Lightweight Directory Access Protocol (LDAP) server in an automated fashion. A configuration tool sub-block 470 includes software for configuring the digitizing block 422. This sub-block 470 optionally includes various

configuration utilities. Commercially available software such as the HP Digital Sender Module includes some features of the exemplary digitizing block 422.

Referring to Fig. 7, an exemplary method for software revision 700 is shown. In a development block 710, a developer (human and/or machine) develops and/or revises a software component for a particular part of a multifunctional device. Next, in a qualification block 714, the software component is qualified (by human and/or machine) in conjunction with another software component (or components) associated with another part (or parts) of the multifunctional device. In the qualification block 714, the revised (or new) software component is combined (by human and/or machine) with at least one other software component to form a software bundle. After qualification, in a placement block 718, the software bundle is placed and/or otherwise made available (e.g., to end-users, administrators, and/or directly to devices). For example, the placement block 718 involves placing the software bundle on a server. Placing optionally includes loading; however, placing may also include linking. In a download block 722, an end-user, administrator, and/or device downloads the software bundle. This exemplary method 700 can eliminate the need to download (or load) individual software components of questionable compatibility. The bundled approach of this exemplary method 700 lends an added assurance of compatibility.

Referring to Fig. 8, an exemplary method for software revision 800 is shown. In a development block (e.g. block 710 of Fig. 7), a developer (human and/or machine) develops a software component 802, such as, a software component for executing a printing operation. Next, in a qualification block (e.g., block 714 of Fig. 7), the software component 802 is qualified (by human

and/or machine) along with additional software components 804, 806. The additional components 804, 806 optionally include components for executing operations related to print media and/or finishing. The software components 802, 804, 806 are bundled (e.g., in a qualification and/or separate bundling block) to form a software bundle 808, which typically exists as a single file. In a placement block (e.g., block 718 of Fig. 7), the software bundle 808 is placed and/or otherwise made available on a server 820. For example, the placement block (e.g., block 718 of Fig. 7) involves placing the software bundle 808 on the server 820. Placing optionally includes loading; however, placing may also include linking.

In this exemplary method 800, a download block (e.g., block 722 of Fig. 7) associated with a multifunctional printer 840 downloads the software bundle 808. Upon downloading, e.g., via a network 830, the software bundle 808 resides in printer memory (see, e.g., printer 110 of Fig. 2). As shown in Fig. 8, the multifunctional printer 840 includes a functional printing block 842, a functional print media block 844 and a functional print finishing block 846. In this example, these three blocks have associated printing, print media and print finishing parts. Once the software bundle 808 resides in printer memory, a processor and associated software may aid in the unbundling of the various software components 802, 804, 806 (see, e.g., printer 110 of Fig. 2). The unbundled components 802, 804, 806 then work in conjunction with the appropriate functional parts.

More specifically, in the exemplary method 800 (shown in Fig. 8), the printing software component may include revised software for an event related to "ready" and/or "offline" status, the print media software component may

include revised software for an event related to the size of a print media tray, and the print finishing software component may include revised software for an event related to a staple supply.

In addition, the software bundle 808 optionally comprises a single file having a specific file extension (e.g., ".RFU"). Upon downloading of a software bundle 808, in the case of pending print jobs or other tasks, the printer may suspend such tasks or allow for execution of tasks prior to initiating revision to printer software.

An exemplary multifunctional printer and/or multifunctional device suitable for use with various exemplary methods, includes an input (e.g., network interface, disk drive, etc.) for receiving a software bundle (e.g., having a specific file extension) wherein the software bundle includes software components for at least two printer/device parts and a processor configured (e.g., through software) to distinguish each of the software components included in the software bundle. Such a processor may also adjust task and/or job priorities to upgrade (or revise) software at a suitable time.

In yet another exemplary method of revising printer software, an administrator initiates execution of Web administration software (e.g., HP WEB JETADMIN®). Next, the administrator enters an IP address or IP hostname of a printer in a device find field. Once found, the administration software displays a status window for the printer. In conjunction with the status window, the administrative software also displays an "update" icon or button. Activation of the update button results in a display of further information related to the type of update to perform (e.g., update printers). Selection of an

"update printers" option causes the administrative software to display an icon or button for uploading of new (or revised) printer software and a browse option to aid in locating an already downloaded software bundle, typically downloaded from a printer manufacturer's Web site. Activation of the icon or button for uploading new (or revised) printer software moves or copies the software bundle from a local storage site to a Web server (e.g., a HP WEB JETADMIN® server). Next, the administrator may refresh the Web browser and select the software bundle file from a menu or the like. Activation of an update software icon or button then causes the administrative software to send the software bundle to a printer needing new or revised software.

Another exemplary method includes using administrative software to revise printer software for a plurality of printers. According to this method, an administrator initiates execution of administrative software (e.g., HP WEB JETADMIN®). Next, the administrator creates a device group, in part, by selecting printers from a list for inclusion in the group. The administrator may update software for printers in this group by activating an "update" icon or button displayed by the administrative software and optionally selecting all or only some of the printers in the group. Exemplary methods described herein, optionally display, send and/or print (automatically and/or manually) information related to a software revision. For example, a printer may transmit (display, send, print, etc.) configuration information verifying a successful revision. In the case of a software failure, a printer may optionally use a stored ship-time version of printer software. Such a stored version may be stored, for example, on a backup partition of a printer's flash DIMM.

In general, the elapsed time for a revision may depend on an I/O transfer time as well as the time that it takes for a printer to reinitialize, which may be required in response to software revision (reinitialization may occur automatically or manually). The I/O transfer time normally depends on factors such as speed of server (or host) making bundle available and the I/O method (fast infrared, parallel, network, etc.). Reinitialization time may depend on factors such as the number of EIO devices installed, the presence of paper handling devices, and the amount of memory installed in the printer.

While various methods described herein refer to multifunctional printers, such methods also optionally apply to multifunctional devices. Various operations described with reference to the exemplary methods are generally performed by a human and/or a machine.

Although various exemplary methods and/or systems have been described in language specific to structural features and/or methodological blocks, it is to be understood that the content of the appended claims is not necessarily limited to the specific features or blocks described. Rather, the specific features and blocks are disclosed as preferred forms of implementing the claimed content.